(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) **EP 2 624 163 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
07.08.2013 Bulletin 2013/32

(51) Int Cl.:
G06F 21/56 *(2013.01)*

(21) Application number: **13153123.8**

(22) Date of filing: **29.01.2013**

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME

(30) Priority: **31.01.2012 US 201213362963**

(71) Applicant: **Trusteer Ltd.**
67442 Tel Aviv (IL)

(72) Inventors:
- **Klein, Amit**
  Herzliya 46433 (IL)
- **Ben-Haim, Eldan**
  Kiryat Ono 55654 (IL)
- **Frishman, Gal**
  Netanya 42309 (IL)

(74) Representative: **Lampis, Marco et al**
Dragotti & Associati Srl
Via Nino Bixio, 7
20129 Milano (IT)

(54) **Method for detecting malware**

(57) System and method for determining, by a security application, whether an examined software code is a malware, according to which the system detects whenever the examined process code performs system calls and further detects a call site. Pieces of code in the surrounding area of the site and/or in branches related to the site are analyzed and the properties of the analyzed pieces of code are compared with a predefined software code patterns, for determining whether the examined process code corresponds to one of the predefined software code patterns. Then the examined process code is classified according to the comparison results.

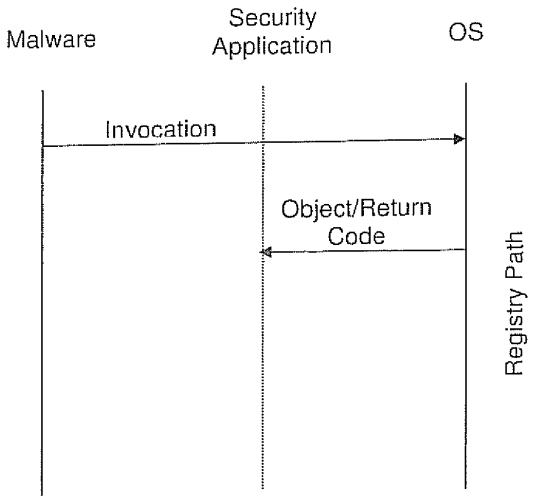

Fig. 2

EP 2 624 163 A1

Description

Field of the Invention

[0001] The present invention relates to the field of Internet security. More particularly, the invention relates to a method for providing more secure browsing and preventing the theft of online sensitive information.

Background of the invention

[0002] As the web browser is becoming the most frequently used application on a personal computer, and as more user confidential data is being entered through the web browser, such as banking and shopping transactions, malicious attacks are being increasingly focused on the web browser. There is an increasing number of malicious exploits that can install malicious code, such that a malicious browser extension persists on a target computer system. For a malicious browser extension to persist on a computer system, typically a malicious file is created so that the malicious extension persists on the disk, and a registry entry associated with the malicious browser extension is created to notify the web browser that a browser extension has been registered with the operating system.

[0003] Thus, for example, if a user enters user confidential data into a form field of a web page, and a malicious browser extension is present on the web browser, when the malicious browser extension receives an event, the malicious browser extension potentially has the ability to access and modify the content of the event. For example, the malicious browser can copy or modify the user confidential data, such as a bank account routing number in the post data parameter of the event, resulting in compromise of the user confidential data.

[0004] To protect consumers and service providers, many software and hardware solutions have been proposed or developed. These methods in general enhance online authentications, but they implicitly assume that the web browser, the most prevalent tool for online activities, is secure. Also, these solutions are not directed to provide protection against malware installation process.

[0005] Another problem of existing solutions is the fact that in order to classify an *.exe file as malicious or benign, protective tools have to scan its content. Sometimes this process may take more time that requires to a malicious file to install itself. This limitation may result in an unavoidable damage to the infected system.

[0006] To address this problem and to protect users from being exploited while browsing the web, browser infection detection tools are required.

[0007] It is therefore an object of the present invention to provide a system which is capable of detecting behavior associated with a malware.

[0008] It is another object of the present invention to provide a system capable of preventing the complete installation of a malware.

[0009] Other objects and advantages of the invention will become apparent as the description proceeds.

Summary of the Invention

[0010] The present invention is directed to a system and method for determining, by a security application, whether an examined software code is a malware, according to which the system detects whenever the examined process code performs invocations (such as is system calls) and further detects a call site (the site within the examined process code, from which the invocations has been launched). One or more pieces of code in the surrounding area of the site and/or in branches related to the site are analyzed and at least a part of the properties of the analyzed pieces of code are compared with a predefined software code patterns, for determining whether the examined process code corresponds to one of the predefined software code patterns. Then the examined process code is classified according to the comparison results.

[0011] The examined process code may be classified as malicious or benign and may be loaded from an executable or from a DLL.

[0012] The examined process code may be a portion of the functions call tree, a specific subroutine or a combination of specific code lines that are spaced from each other by several code instructions.

[0013] The security application or at least a portion thereof, may reside on the client device.

[0014] Detection of attempts to install a malware may be made offline or in real-time. Upon detection the malware is prevented from completing its installation.

[0015] In one aspect, the security application that may be called from an application or from the operating system compares each analyzed piece of software code against a "black list" of known malware patterns.

Brief Description of the Drawings

[0016] In the drawings:

- Fig. 1 is a flow chart generally illustrating the method of the invention.
- Fig. 2 is a flow chart generally illustrating an embodiment of the invention.

Detailed Description of Preferred Embodiments

[0017] The figures and the following description relate to embodiments of the present invention by way of illustration only. It should be noted that from the following discussion, alternative embodiments of the structures and methods disclosed herein will be readily recognized as viable alternatives that may be employed without departing from the principles of the claimed invention.

[0018] Reference will now be made to several embodiments of the present invention(s), examples of which are

illustrated in the accompanying figures. Wherever practicable similar or like reference numbers may be used in the figures and may indicate similar or like functionality. The figures depict embodiments of the present invention for purposes of illustration only. One skilled in the art will readily recognize from the following description that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention described herein.

[0019] Unless otherwise indicated, the functions described herein may be performed by executable code and instructions stored in computer readable medium and running on one or more processor-based systems. However, state machines, and/or hardwired electronic circuits can also be utilized. Further, with respect to the example processes described herein, not all the process states need to be reached, nor do the states have to be performed in the illustrated order. Further, certain process states that are illustrated as being serially performed can be performed in parallel.

[0020] Similarly, while certain examples may refer to a Personal Computer (PC) system, Mac, other computer or electronic systems can be used as well, such as, without limitation, a network-enabled personal digital assistant (PDA), a smart phone, a tablet, and so on.

[0021] The present invention relates to a method for real-time detection of attempts to install a malware in a computer system (e.g., in order to infect web browsers with malware). According to an embodiment of the invention, and as will be exemplified hereinafter, a method is provided for preventing from a malware (or other suspicious software code) to complete its installation.

[0022] The term "malware" refers herein to a malicious code that is defined as any computer program, module, set of modules, or code that enters a computer system environment without an authorized user's knowledge and/or without an authorized user's consent. A malicious browser extension, such as a malicious BHO, that can access user private data in a browser event is one example of malicious code. Further herein, malicious activity is any activity resulting from the execution of a malicious code.

[0023] In the present embodiment, the web browser is any one of a number of conventional web browser applications, such as Microsoft Internet Explorer, or Firefox. Embodiments in accordance with the invention detect and may prevent attempts to install malwares. In one embodiment, a security application is installed on a host computer system that is registered to detect invocations from the OS of the host computer. Such invocations may be function calls that interact with the Windows Registry of the host computer (in case where the OS is Microsoft Windows) or with file system operations.

[0024] When an invocation is detected, a determination is made whether the properties of program code from which that invocation was launched represent a malware. In one embodiment, if the invocation is not determined to be a suspicious event, the event is enabled for further processing (e.g., for completing the installation process). Alternatively, if the invocation is determined to be a malware, the installation process of that software code is terminated.

[0025] Referring now to Fig. 1, a diagram of a computer system including a security application for detecting attempts to install and execute malwares on a host computer system is shown in accordance with an embodiment of the present invention. The host computer system, sometimes called a user device, typically includes a central processing unit (CPU), an input output (I/O) interface, and a memory, including an operating system and a web browser. In one embodiment, operating system maintains an invocations event order list that provides the site from which invocations events are received.

[0026] In one embodiment, the memory includes the security application, which comprises two main components: a) a detection engine that is configured to monitor invocations in the registry of the Operation System (OS) by listening to the set registry path of the OS, and b) a database preloaded with predefined parameters which represent the behavior of selected pieces of software codes of different malwares.

[0027] According to an embodiment of the present invention, the security application performs the following tasks: At first it detects whenever a software code performs an invocation activity (e.g., a system call). At the next step, it detects the site within the invoking code, from which the invocation has been launched. This is done in order to analyze one or more pieces of code (that may belong to an executable or to a DLL) in the surrounding area of that site and/or in branches related to that site that are responsible for launching the invocation. For example, the piece of code to be analyzed may be associated with portions of the functions call tree (that shows the hierarchy of function calls in a program), such as a specific subroutine or a combination of specific code lines that are spaced from each other by several code instructions. This results in a subset of properties in the form of a unique pattern, which can then be associated with the existence of malware and even with the type of malware.

[0028] At the next step it checks whether the properties of the analyzed pieces of codes (or at least part of them) matches the predefined software code parameters or patterns stored in the database. A match between the properties of the analyzed pieces of codes and the predefined software code indicates that the software is a malware or any other type of software code that is needed to be detected. This may be done by comparing the properties of the analyzed pieces of code with predefined software code patterns and determining whether the examined process code corresponds to one of the predefined software code patterns. Then, the examined process code is classified according to the comparison results.

[0029] For example, known behavior of several types of malwares such as SpyEye is to first add an entry in **HKCU\Software\Microsoft\Windows\CurrentVersion\Run Windows Registy key** in order to load itself

at the startup sequence of Windows (when the user logs in).

[0030] Typically when the OS receives an invocation, it transfers the system request through various layers. As described, should a malware code tries to permanently install itself by adding a value to a Registry Key of the Windows Registry, the security application monitors this attempt and detects the site from which said invocations has been launched and pieces of code in the surrounding area of this site.

[0031] In one embodiment, the security application compares each analyzed piece of software code against a "black" list of known malware patterns. When one or more pieces of codes matches an entry in the "black" list, the associated software code that tries to install itself is terminated or blocked; otherwise, the software code is allowed to complete the installation.

[0032] Referring now to Fig. 2, illustrates a diagram representation of the notification order list in the registry path of the OS at the first stages of a malware installation attempt in accordance with an embodiment of the invention.

[0033] As used herein, a computer memory refers to a volatile memory, a non-volatile memory, or a combination of the two. Although the security application is referred to as an application, this is illustrative only. The security application should be capable of being called from an application or the operating system. In one embodiment, an application is generally defined to be any executable code. Moreover, those of skill in the art will understand that when it is said that an application or an operation takes some action, the action is the result of executing one or more instructions by a processor.

[0034] As illustrated in Fig. 1, since the security application is notified on OS-level system calls, this medium or at least a portion thereof, resides on the computer system itself (client device) and the remaining portion may be stored in a memory that is physically located in a location different from the host computer. This could be accomplished in a client-server system, or alternatively via a connection to another computer via modems and analog lines, or digital interfaces and a digital carrier line.

[0035] In view of this disclosure, the functionalities of the security application in accordance with the embodiments of the present invention can be implemented in a wide variety of computer system configurations. In addition, the functionalities of the security application could be stored as different modules in memories of different devices. For example, security the application could initially be stored in computer system, and then as necessary, a portion of the security application could be transferred to the host computer system and executed on the host computer system. Consequently, part of the functionality of the security application would be executed on the processor of server computer system, and another part would be executed on processor of the host computer system.

[0036] In view of this disclosure, those of skill in the art can implement various embodiments of the present invention in a wide-variety of physical hardware configurations using an operating system and computer programming language of interest to the user. In yet another embodiment, the security application is stored in a memory of a server computer system. The security application is transferred over a network to the memory in a host computer system.

[0037] While some embodiments of the invention have been described by way of illustration, it will be apparent that the invention can be carried into practice with many modifications, variations and adaptations, and with the use of numerous equivalents or alternative solutions that are within the scope of persons skilled in the art, without departing from the spirit of the invention or exceeding the scope of the claims.

Claims

1. A method for determining, by a security application, whether an examined software code is a malware, comprising the steps of:

   a. detecting whenever said examined process code performs invocations;
   b. detecting a call site being the site within said examined process code, from which said invocations has been launched;
   c. analyzing one or more pieces of code in the surrounding area of said site and/or in branches related to said site; and
   d. comparing at least a part of the properties of the analyzed pieces of code with a predefined software code patterns and determining whether said examined process code corresponds to one of said predefined software code patterns; and
   e. classifying said examined process code according to the comparison results.

2. A method according to claim 1, wherein the process code is classified as malicious or benign.

3. A method according to claim 1, wherein the examined process code loaded from an executable or from a DLL.

4. A method according to claim 1, wherein the examined process code is a portion of the functions call tree, a specific subroutine or a combination of specific code lines that are spaced from each other by several code instructions.

5. A method according to claim 1, wherein the security application or at least a portion thereof, resides on the client device.

6. A method according to claim 1, wherein detection of attempts to install a malware is made in real-time.

7. A method according to claim 6, wherein upon detection the malware is prevented from completing its installation.

8. A method according to claim 1, wherein the invocation activity is a system call.

9. A method according to claim 1, wherein the security application compares each analyzed piece of software code against a "black list" of known malware patterns.

10. A method according to claim 1, wherein the security application is capable of being called from an application or from the operating system.

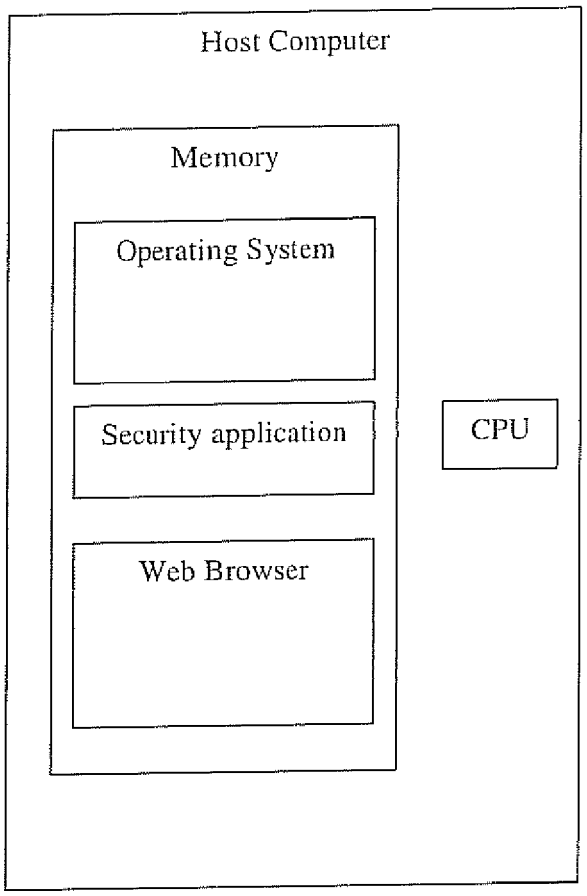
Host Computer
Memory
Operating System
Security application
Web Browser
CPU

Fig. 1

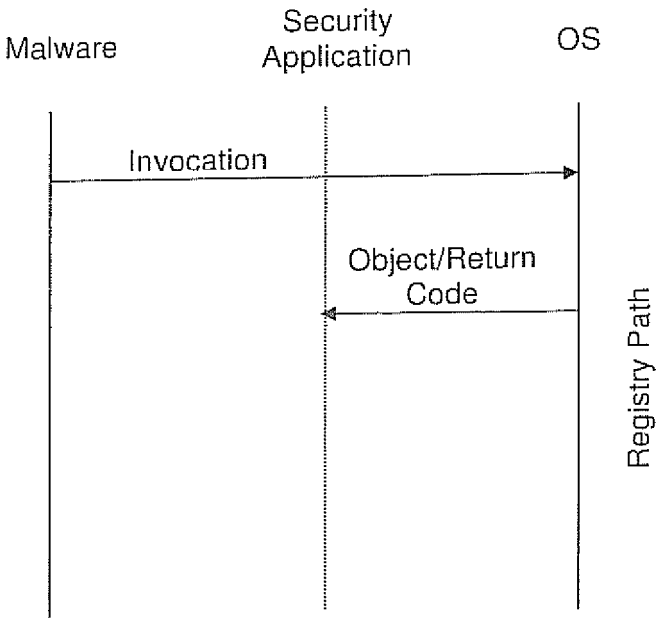
Malware
Security Application
OS
Invocation
Object/Return Code
Registry Path

Fig. 2

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 13 15 3123

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	US 7 340 777 B1 (SZOR PETER [US]) 4 March 2008 (2008-03-04) * column 4 - column 9; figures 2,4 * -----	1-10	INV. G06F21/56
A	US 2008/052679 A1 (BURTSCHER MICHAEL [US]) 28 February 2008 (2008-02-28) * paragraph [0034] - paragraph [0037] * -----	4	
A	US 2010/107257 A1 (OLLMANN GUNTER [US]) 29 April 2010 (2010-04-29) * paragraph [0003] * * paragraph [0021] * -----	5	
			TECHNICAL FIELDS SEARCHED (IPC) G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	20 March 2013	Chabot, Pedro

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 13 15 3123

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

20-03-2013

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7340777 B1	04-03-2008	NONE	
US 2008052679 A1	28-02-2008	NONE	
US 2010107257 A1	29-04-2010	CA 2719495 A1	06-05-2010
		CN 102171987 A	31-08-2011
		EP 2294786 A2	16-03-2011
		JP 2012507094 A	22-03-2012
		KR 20110076976 A	06-07-2011
		US 2010107257 A1	29-04-2010
		US 2012084862 A1	05-04-2012
		WO 2010049273 A2	06-05-2010

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82